# JavaScript ES6 Concepts

## ES6 Features (ECMAScript 2015)

let and const keywords for block-scoped variable declaration.

Arrow functions providing concise syntax and lexical `this` binding.

Classes offering syntactic sugar over JavaScript prototypes.

Template literals using backticks (`) for string interpolation.

Default parameters in functions.

Destructuring assignment for arrays and objects.

Rest and spread operators (`...`).

Modules using import and export.

Promises for asynchronous programming.

Enhanced object literals with shorthand and computed property names.

Map and Set data structures.

## JavaScript `let`

Introduced in ES6.

Declares a block-scoped variable.

Not hoisted like var.

Prevents redeclaration in the same scope.

Example:  
let name = "John";  
if (true) {  
 let name = "Doe";  
 console.log(name); // "Doe"  
}  
console.log(name); // "John"

## Difference Between `var` and `let`

| Feature | var | let |

|------------------|-------------------------------|-------------------------------|

| Scope | Function-scoped | Block-scoped |

| Hoisting | Yes (initialized as undefined)| Yes (but not initialized) |

| Redeclaration | Allowed | Not allowed in the same scope |

| Temporal Dead Zone| No | Yes |

## JavaScript `const`

Declares a constant (block-scoped).

Must be initialized at the time of declaration.

Cannot be reassigned, but the value (like object properties) can be modified.

Example:  
const person = { name: "Alice" };  
person.name = "Bob"; // Allowed  
// person = {}; // ❌ Error

## ES6 Class Fundamentals

ES6 introduced class syntax to simplify object-oriented programming.

A class defines a blueprint for objects.

Example:  
class Animal {  
 constructor(name) {  
 this.name = name;  
 }  
  
 speak() {  
 console.log(`${this.name} makes a sound`);  
 }  
}

## ES6 Class Inheritance

Use `extends` to inherit from another class.

Use `super()` to call the parent class constructor.

Example:  
class Dog extends Animal {  
 speak() {  
 console.log(`${this.name} barks`);  
 }  
}  
  
const d = new Dog("Buddy");  
d.speak(); // Buddy barks

## Arrow Functions

Introduced in ES6 with shorter syntax.

Do not bind their own `this`.

Syntax:  
const add = (a, b) => a + b;  
const sayHello = () => console.log("Hello");

## Set() and Map()

Set(): Collection of unique values.  
Example:  
let s = new Set([1, 2, 2, 3]);  
console.log(s); // Set(3) {1, 2, 3}

Map(): Key-value pairs.  
Example:  
let m = new Map();  
m.set("name", "Alice");  
m.set(1, "one");  
console.log(m.get(1)); // "one"